

TP 2 — Création d'un design en CSS

Dans ce TP, nous allons reprendre la page HTML créée lors du TP 1 et ajouter un fichier CSS pour créer le design. Vous pourrez vous inspirer du travail réalisé lors du TD 2 pour retrouver les propriétés et les valeurs CSS dont vous aurez besoin. L'objectif est de reproduire le plus fidèlement possible l'apparence de la capture d'écran donnée sur EPREL.

À rendre

Vous devrez rendre l'ensemble des fichiers HTML/CSS produits et les images utilisées dans les pages.

Vous placerez l'ensemble des fichiers produits dans une archive ZIP nommée `r102_tp2_<nomDeFamille>.zip` et vous déposerez cette archive sur EPREL.

1 Préparatifs

- 1) Placez-vous dans votre répertoire `Documents` ▶ `BUT_INFO` ▶ `R102_DIW` et créez un sous-dossier `TP2`.
- 2) Faites un copier/coller de `Documents` ▶ `BUT_INFO` ▶ `R102_DIW` ▶ `TP1` dans le dossier nouvellement créé.
- 3) À partir de maintenant, nous considérerons que le répertoire `Documents` ▶ `BUT_INFO` ▶ `R102_DIW` ▶ `TP2` est la racine du projet. Ouvrez `QuickPHP` et configurez-le correctement.
- 4) Créez un dossier `html` à la racine et déplacez-y le fichier `index.html`. Vous avez maintenant l'arborescence suivante :

```
1 /
2   css/
3     design.css
4   html/
5     index.html
6   img/
7     portrait1.jpg
8     portrait2.jpg
```

Arborescence

Mettez à jour l'en-tête du fichier HTML pour le lier au fichier CSS.

Dans la suite de ce TP, on ne modifiera plus le fichier HTML, sauf en cas d'indication contraire.

L'énoncé fonctionne comme un cahier des charges : il décrit les spécifications du design attendu. Votre travail consiste à retranscrire en CSS ces instructions. Afin de vous y aider, des captures d'écran des résultats souhaités sont fournies.

2 Au travail !

2.1 Configuration de la page

- 1) La largeur de la page est fixée à 800 pixels. On lui attribue un arrière-plan gris, de code hexadécimal `f7f7f7`. La police d'écriture sera Verdana, puis serif en deuxième choix. Enfin, le contenu devra être au milieu de l'écran.

2.2 Mise en forme des titres

- 2) Titre de niveau 1.
 - a. Le titre s'affiche en orange (hexadécimal : `ca6d16`), sur un arrière-plan blanc, en gras (épaisseur 400) et la police est 2,4 fois plus grande que celle de base.

Rappel sur les longueurs en CSS

Il existe des unités de longueur *absolues* et des unités *relatives* en CSS. En longueurs relatives, la taille « de base » de la police est de longueur `1rem` ou `1em`. Changer une taille de police à `2rem` revient à doubler sa taille par rapport à la longueur de référence.

- b. La marge basse est deux fois plus grande que la taille de police de base. À gauche, la marge interne est 60% de la taille de police.
 - c. Enfin, on trace une ligne supérieure de couleur `e5e8eb` de taille 0,2 fois la taille de référence.

Le rendu doit être le suivant :

Grace Hopper

On rappelle que les bordures se gèrent avec le sélecteur CSS `border`. Les marges, avec le sélecteur `margin` et les marges intérieures avec `padding`. Consultez la documentation pour les détails.

- 3) Titres de niveau 2.
 - a. La taille de police est 1,8 fois la taille de référence.
 - b. On applique une marge basse de 1,8 fois la taille de référence.
 - c. La couleur est `ca6d16`.
 - d. Une bordure en pointillés de 1 pixel est utilisée pour souligner le titre.

Voici le résultat attendu (observez les petits points orange en bas du titre) :

Biographie

- 4) Titres de niveau 3.
 - a. La couleur de police choisie est `520d35`.
 - b. La marge au-dessus est fixée à 15 pixels, et la marge inférieure à 5 pixels.
 - c. La taille de police est 1,4 fois la taille de base.

2.3 Mise en forme des paragraphes et liens

- 5) Le texte des paragraphes est justifié et présenté sur deux colonnes (effectuer une recherche pour trouver comment obtenir ce résultat).
- 6) Les liens s'affichent en couleur 1088bf et le soulignement disparaît au passage de la souris.

2.4 Positionnement des images

2.4.1 Les class et les id

Le mécanisme utilisé en HTML/CSS pour distinguer des éléments contenus dans les mêmes balises est celui des classes et des identifiants.

Testez les codes suivants pour mieux comprendre les choses :

	HTML
1	<code><p class="p_italique">Un paragraphe en italique.</p></code>
2	<code><p id="p_gras">Un paragraphe en gras.</p></code>
3	<code><p class="p_italique">Un autre paragraphe en italique.</p></code>
4	<code><p>Un paragraphe sans rien de particulier</p></code>

	CSS
1	<code>p.p_italique {</code>
2	<code> font-style: italic;</code>
3	<code>}</code>
4	<code>p#p_gras {</code>
5	<code> font-weight: bold;</code>
6	<code>}</code>

On ajoute à la balise `<p>` un attribut `class="p_italique"` et dans le fichier CSS, le sélecteur `p.p_italique` s'appliquera uniquement aux paragraphes ayant la classe `p_italique`. Ainsi, on pourra affecter des styles différents à des paragraphes, qui sont pourtant délimités par les mêmes balises `<p>...</p>`.

L'attribut `id` fonctionne de la même manière, sauf qu'un `id` (comme son nom l'indique) ne peut être utilisé *qu'une seule et unique fois* pour toute la page. Le sélecteur CSS pour un `id` utilise un dièse (`#`), alors que pour une `class`, c'est un point (`.`).

2.4.2 Mise en pratique

Notre site contient deux images : `img▶portrait1.jpg` et `img▶portrait2.jpg`. Nous souhaitons que ces deux images s'affichent différemment :

- Pour le premier portrait, il doit être centré (horizontalement) et occuper au plus la moitié de la largeur :

Grace Murray Hopper est une informaticienne américaine et *Rear admiral (lower half)* de la marine américaine, née le 9 décembre 1906 à New York et morte le 1er janvier 1992 dans le comté d'Arlington. Elle est la conceptrice du premier compilateur en 1951 (A-0 System) et du langage Cobol en 1959.



- La deuxième image doit être flottante et être entourée par du texte. Notez par ailleurs que le paragraphe qui encadre l'image est maintenant sur une seule colonne au lieu de deux. Une marge intérieure de 15 pixels est appliquée en haut et à droite de l'image.

Carrière

En 1943, Grace Hopper s'engage dans la marine américaine. Promue au grade de lieutenant, elle est affectée l'année suivante au *Bureau of Ordnance Computation Project* de [l'université Harvard](#). Elle travaille dans l'équipe de Howard Aiken sur l'ordinateur Harvard Mark I. Elle fait partie du premier groupe, comptant trois personnes, apprenant à le programmer.



À la fin de la Seconde Guerre mondiale, elle quitte le service actif de la marine, mais continue à travailler au développement des ordinateurs Harvard Mark II puis Harvard Mark III pour le compte du Computation Laboratory d'Harvard.

Nous allons donc avoir besoin de distinguer les deux images, car nous souhaitons appliquer deux styles différents à des éléments délimités par les mêmes balises `<img>`. Il en va de même pour le paragraphe que l'on doit présenter sur seule colonne au lieu de deux : il faut le différencier des autres paragraphes du site.

Pour les deux prochaines questions, vous devrez apporter des modifications au code HTML.

- 7) En utilisant convenablement des attributs HTML `class` et `id`, mettez en place les spécifications demandées pour la deuxième image.
- 8) La deuxième image doit être centrée horizontalement et occuper au plus la moitié de la largeur disponible. Vous aurez besoin de modifier cette image pour en faire un élément de type `block`.

Les boîtes en CSS et le *responsive design*

La gestion des boîtes en CSS est un sujet complexe et délicat. En particulier car il faut assurer la compatibilité du design avec différentes tailles d'écran... voire même avec des supports différents, notamment mobile.

Les designs compatibles avec les supports variés sont qualifiés de *responsive*. Cela utilise notamment la propriété CSS `flex`, qui permet de créer des boîtes élastiques, de taille variable. Cette notion ne sera pas abordée dans le cadre de cette ressource introductive. Consultez [cette page](#) pour en savoir plus.

3 Soyez créatif·ves

3.1 Un aparté

Il peut arriver que nous ayons besoin de créer des boîtes, mais qu'il n'existe aucune balise HTML ayant la sémantique correspondante. Cette situation est en principe assez rare, mais il existe pour y remédier une balise « neutre », qui crée une boîte sans aucune signification particulière :

```
1 <p>
2   Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc convallis lacinia
3   mauris, vitae laoreet ex malesuada non. Nunc porta congue [...].
4   <div class="div_exemple"> <!-- Balise neutre -->
5       Morbi scelerisque ipsum a tellus rutrum bibendum. Sed tincidunt massa ac [...].
6   </div>
7 </p>
```

```
1 p {
2   border: 1px solid red;
3 }
4
5 p .div_exemple { /* Style appliqué au sélecteur de classe div_exemple dans un <p> */
6   border: 1px solid blue
7 }
```

N'hésitez pas à tester et modifier ce code pour comprendre son effet. Il existe un analogue de la balise `<div>...</div>` pour les éléments de type inline. Il s'agit de `<span class="span_exemple">...</span>`, qui fonctionne de manière identique.

Du bon usage des `div` et des `span`

Les `div` et les `span` sont très souvent utilisées alors qu'une balise HTML sémantique existe pour remplir le même rôle. *Il s'agit d'une (très) mauvaise pratique.*

Les balises `div` et `span` doivent être utilisées en dernier recours, si aucune balise HTML n'existe pour désigner la signification de l'élément.

3.2 Deuxième page du site

- 9) Créez une deuxième page HTML, dans laquelle vous relaterez la biographie d'un·informaticien·ne de votre choix. Vous utiliserez le même fichier CSS pour les deux pages.
- 10) Ajoutez, dans les deux pages, un menu avec les balises `<nav>...</nav>`, contenant les liens d'une page à l'autre.
- 11) À l'aide de CSS, modifiez à votre convenance le style du menu que vous venez de créer.

4 Pour terminer...

- 12) Passez vos codes HTML et CSS aux validateurs et corrigez toutes les erreurs et warnings.

Quelques bonnes pratiques

- Chaque type de fichiers doit être placé dans un répertoire spécifique (images, vidéos, HTML, CSS, etc.).
- Les codes HTML et CSS doivent être valides.
- On utilise les attributs `class` et `id` avec parcimonie et uniquement si on n'a pas d'autre solution. Les noms des classes et des identifiants doivent être choisis avec soin pour maintenir un code lisible.